Logic (PHIL 2080, COMP 2620, COMP 6262)

Chapter: Introduction to Logic

Pascal Bercher

AI Group
School of Computing
College of Engineering and Computer Science
the Australian National University

21 & 22 February 2022

Organizational Matters

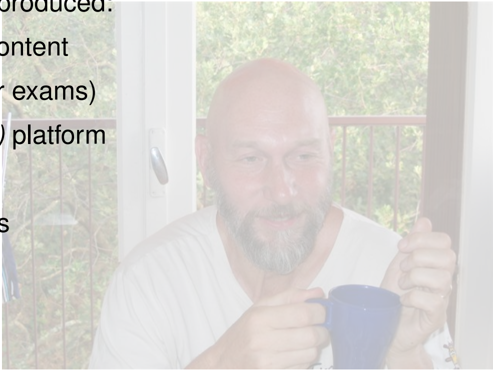
Team: Convenors & Lecturers, 1/3

Prof. Dr. John Slaney (2011 – 2020, now retired)

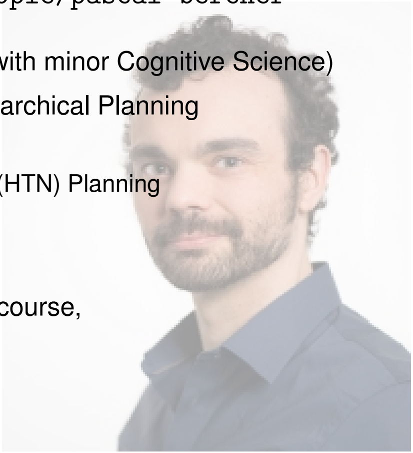
`http://users.cecs.anu.edu.au/~jks/`

We inherited his course; he produced:

- Course structure and content
- Most exercises (also for exams)
- The *Logic for Fun (L4F)* platform
- Its plagiarism scanner
- The online course notes



Team: Convenors & Lecturers, 2/3



Dr. Pascal Bercher since 2021

`https://cecs.anu.edu.au/people/pascal-bercher`

- *Studies:* Computer Science (with minor Cognitive Science)
- *PhD:* Computer Science: Hierarchical Planning
- *Research:*
  - Hierarchical Task Network (HTN) Planning
  - Heuristic Search
  - Complexity Theory

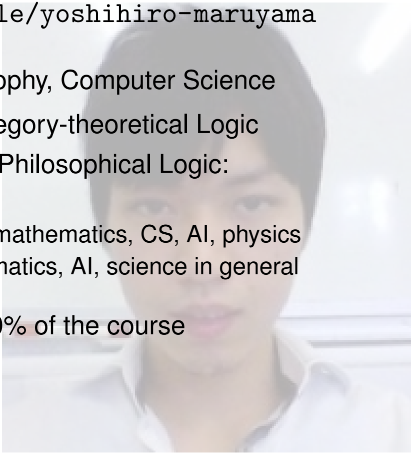
→ Pascal is the convenor of the course,

→ and teaches the first 50%.

Team: Convenors & Lecturers, 3/3

Dr. Yoshihiro Maruyama since 2021

`https://cs.anu.edu.au/people/yoshihiro-maruyama`



- *Studies:* Mathematics, Philosophy, Computer Science
- *PhD:* Computer Science: Category-theoretical Logic
- *Research:* Mathematical and Philosophical Logic:
  - category-theoretical logic
  - categorical foundations of mathematics, CS, AI, physics
  - philosophy of logic, mathematics, AI, science in general

→ Yoshi will teach the second 50% of the course

Team: The Tutors

- See Wattle for the complete list and contact info.
  They also provided a short introduction in the forum.
- What do they do?
  - Give the tutorials/workshops
  - Answer your questions (via Wattle forum)
  - (Co-)Mark the homework, assignments, and exam

Various: Appointments/Dates

- Lectures:
  - Online live, recordings available via Wattle/Echo360
  - 2 per week, (approx.) 60 minutes each, Mondays & Tuesdays

Various: Appointments/Dates

- Lectures:
  - Online live, recordings available via Wattle/Echo360
  - 2 per week, (approx.) 60 minutes each, Mondays & Tuesdays
- Tutorials:
  - Once per week, 2 hours, in-person or zoom for those who are not on campus.
  - Self-enrollment starts and ends this week. But first take part in the survey!
  - We'll do both tutorial-like "standard" exercises as well as workshop-like modeling tasks

Various: Appointments/Dates

- Lectures:
  - Online live, recordings available via Wattle/Echo360
  - 2 per week, (approx.) 60 minutes each, Mondays & Tuesdays
- Tutorials:
  - Once per week, 2 hours, in-person or zoom for those who are not on campus.
  - Self-enrollment starts and ends this week. But first take part in the survey!
  - We'll do both tutorial-like "standard" exercises as well as workshop-like modeling tasks
- Appointments:
  - Only in exceptional cases. If required ask for appointment via mail
  - Otherwise ask all questions via the forum or your tutor

Various: Appointments/Dates

- Lectures:
  - Online live, recordings available via Wattle/Echo360
  - 2 per week, (approx.) 60 minutes each, Mondays & Tuesdays
- Tutorials:
  - Once per week, 2 hours, in-person or zoom for those who are not on campus.
  - Self-enrollment starts and ends this week. But first take part in the survey!
  - We'll do both tutorial-like "standard" exercises as well as workshop-like modeling tasks
- Appointments:
  - Only in exceptional cases. If required ask for appointment via mail
  - Otherwise ask all questions via the forum or your tutor
- Drop-in Sessions:
  - You can ask your questions or just listen in. Intuitive explanations!
  - Date to be decided. Take part in the (second) survey!

Various: Exercises and Exam

- Homework (mostly) each week:
  - Standard exercises (do proofs) or modeling tasks
  - Get corrected by tutors, marks are just FYI, they do *not* count towards the exam/course mark
  - Collaboration (up to 3 people) is strongly encouraged, but please don't hand in the same results several times

Various: Exercises and Exam

- Homework (mostly) each week:
  - Standard exercises (do proofs) or modeling tasks
  - Get corrected by tutors, marks are just FYI, they do *not* count towards the exam/course mark
  - Collaboration (up to 3 people) is strongly encouraged, but please don't hand in the same results several times
- Three Assignments:
  - 1 related to formal proofs, 1 to modeling, and 1 essay
  - Each assignment counts 15% of the final mark
  - Any form of cheating will be escalated and has serious consequences. We use software!
  - Deadlines: Are strict, no exceptions (unless you have a *serious* reason, backed up by medical certificates, for example)

Various: Exercises and Exam

- Homework (mostly) each week:
  - Standard exercises (do proofs) or modeling tasks
  - Get corrected by tutors, marks are just FYI, they do *not* count towards the exam/course mark
  - Collaboration (up to 3 people) is strongly encouraged, but please don't hand in the same results several times
- Three Assignments:
  - 1 related to formal proofs, 1 to modeling, and 1 essay
  - Each assignment counts 15% of the final mark
  - Any form of cheating will be escalated and has serious consequences. We use software!
  - Deadlines: Are strict, no exceptions (unless you have a *serious* reason, backed up by medical certificates, for example)
- Exam
  - Will be online, (likely) uses proctorio, 3 hours
  - Counts 55% of final mark

Various: Course Material

- Slides (see Wattle)
- Online book “Logic Notes” (`http://users.cecs.anu.edu.au/~jks/LogicNotes/index.html`)
- Our modeling tool “Logic for Fun (L4F)” (`https://l4f.cecs.anu.edu.au/`)
  Currently offline, we are working on it!
- Online forum! (Set Wattle reminders accordingly!)
  Please read the rules! (Search first, use descriptive titles, etc.)
- For further reading, see books:
  - G. Restall. *Logic: An Introduction*. Ed. by J. Shand. Routledge, 2005 (Well-suited for Philosophy students)
  - D. van Dalen. *Logic and Structure*. Springer, 2012 (Well-suited for Computer Science and Mathematics students)

Various: Feedback/Corrections

- Nobody is perfect!
- Did you find an error in the slides? (Even just a typo!)
- Do you have an idea on how to improve the slides?
  - More content? Less content?
  - Adding a specific example?
  - Adding a specific explanation?
  - Explaining a specific error students typically make?

→ Let us know! Drop the convenor, lecturer, or course representative an email!

Various: Course Representatives

- Each course code (COMP2620, COMP6262, PHIL2080) has two course representatives
- Their job is:
  - to act as the *official* liaison between your peers and convener
  - you can conduct survey about the course and influence it by feeding back the results to the convenor/lecturer. (Note that there are two lecturers!)
  - See slides on Wattle or the pre-recorded lecture (for these slides)

→ Interested? Nominate yourself! Drop Pascal an email:

  - Note the deadline: March 2nd!
  - Name the course code you are nominating yourself for
  - Elaborate your motivation for doing so

Motivation

Philosophy and Computer Science: Philosophy

- Logic is the science of representing and reasoning about knowledge.
- Reasoning about what follows from some knowledge (base) is *clearly* an important question!
  - Cogito, ergo sum (Latin)
  - Ich denke, also bin ich (German)
  - I think, therefore I am (English)
- A more detailed motivation (and history!) of Logic can be found in Yoshi's (13 minute) presentation available in Echo360.

Philosophy and Computer Science: Computer Science

- Computer hardware bases on logic gates: NOT, AND, OR, XOR etc. So *all* computers’ hardware is based on logic.

Philosophy and Computer Science: Computer Science

- Computer hardware bases on logic gates: NOT, AND, OR, XOR etc. So *all* computers' hardware is based on logic.
- Logic plays an important role in Theoretical Computer Science (Complexity Theory and more). Many important problems are NP-complete, and the most famous and important problem is SAT (can a logical formula be made true?)

Philosophy and Computer Science: Computer Science

- Computer hardware bases on logic gates: NOT, AND, OR, XOR etc. So *all* computers' hardware is based on logic.
- Logic plays an important role in Theoretical Computer Science (Complexity Theory and more). Many important problems are NP-complete, and the most famous and important problem is SAT (can a logical formula be made true?)
- Many practically relevant problems (also optimization problems) like, e.g., Traveling Salesman can be phrased as SAT problem and thus solved automatically.

Philosophy and Computer Science: Computer Science

- Computer hardware bases on logic gates: NOT, AND, OR, XOR etc. So *all* computers' hardware is based on logic.
- Logic plays an important role in Theoretical Computer Science (Complexity Theory and more). Many important problems are NP-complete, and the most famous and important problem is SAT (can a logical formula be made true?)
- Many practically relevant problems (also optimization problems) like, e.g., Traveling Salesman can be phrased as SAT problem and thus solved automatically.
- *Many* disciplines require/model knowledge of some sort. It is thus modeled via Logic. Think of medical data bases, implemented as ontologies: there are relationships between certain body parts and their functionality, which can be modeled allowing us to make inferences providing certain knowledge (like symptoms or dysfunctional organs).

Making good Arguments: What is Logic?

- Logic is the science of reasoning, i.e., making arguments.
  - Good/correct reasoning vs. bad/wrong reasoning
  - Making (and reasoning about) valid arguments

  ⇒ See (famous) Monty Python sketch “argument clinic” (e.g., `https://www.dailymotion.com/video/x2hwqn9`)

. . .

Person 1: Well, an argument is not the same as contradiction.

Person 2: It can be.

Person 1: No, it can’t.

Person 2: An argument is a connected series of statements to establish a definite proposition.

Person 1: No, it isn’t.

Person 2: Yes, it is!

. . .

Making good Arguments: What's an Argument?

Example:

- All footballers are bipeds

Making good Arguments: What's an Argument?

Example:

- All footballers are bipeds
- Socrates it a footballer

Making good Arguments: What’s an Argument?

Example:

- All footballers are bipeds
- Socrates it a footballer
- Thus, Socrates is a biped

Making good Arguments: What’s an Argument?

Example:

- All footballers are bipeds
- Socrates it a footballer
- Thus, Socrates is a biped

→ This is a valid argument

Making good Arguments: What’s an Argument?

Example:

$$\left.\begin{array}{l}\bullet\ \text{All footballers are bipeds}\\[1em] \bullet\ \text{Socrates it a footballer}\end{array}\right\}\ \text{premises}$$

$$\left.\begin{array}{l}\bullet\ \text{Thus, Socrates is a biped}\end{array}\right\}\ \text{conclusion}$$

→ This is a valid argument

Arguments consist of premises and a conclusion.

Making good Arguments: What’s an Argument?

Another Example:

- All cats are insects
- Snoopy is a cat

premises

- Thus, Snoopy is an insect

conclusion

→ This is also a valid argument!

- Although everything was wrong!
- All premises and the conclusion!

Making good Arguments: What's an Argument?

Another Example:

- All cats are insects
- Snoopy is a cat

} premises

- Thus, Snoopy is an insect } conclusion

→ This is also a valid argument!

- Although everything was wrong!
- All premises and the conclusion!

→ But we don't care, since it has a valid *form*. We exploit this form, and abstract from the content to reason about the conclusions.

Making good Arguments: What's an Argument?

Our final Example:

- All logicians are rational
- Restall[1] is rational

premises

- Thus, Restall is a logician

conclusion

→ Interestingly, this is an invalid (wrong!) argument!

- Although everything was right!
- All premises and the conclusion!

→ *Wrong form*: The conclusion did not *follow* from the premises.

[1] Greg Restall, professor of logic at the University of Melbourne, author of the best-known book on substructural logic and editor in chief of the Australasian Journal of Logic, is presumably a logician if anyone is.

Making good Arguments: Forms of Arguments

Valid arguments have, e.g., the following form:

- All *A*s are *B*s;
- *x* is an *A*;
- Therefore, *x* is a *B*.

Making good Arguments: Forms of Arguments

Valid arguments have, e.g., the following form:

- All *A*s are *B*s;
- *x* is an *A*;
- Therefore, *x* is a *B*.

The example with Restall did not work because it used a wrong form:

- All *A*s are *B*s;
- *x* is an *B*;
- Therefore, *x* is an *A*.

Making good Arguments: Valid Arguments

- An argument is considered valid, whenever the conclusion logically follows from the premises.
- “Logically follows” abstracts away from the number of “intermediate steps” that are required so that the conclusion becomes “obvious”.
- For example, if we take all axioms of some mathematical system as the premises and one of its (valid) theorems/propositions as its conclusion, this forms a valid argument – no matter how ingenious the theorem is!

Making good Arguments: Valid Arguments

- An argument is considered valid, whenever the conclusion logically follows from the premises.
- “Logically follows” abstracts away from the number of “intermediate steps” that are required so that the conclusion becomes “obvious”.
- For example, if we take all axioms of some mathematical system as the premises and one of its (valid) theorems/propositions as its conclusion, this forms a valid argument – no matter how ingenious the theorem is!
- Thus, showing that an argument is actually valid is hard!
- We will break down arguments into a sequence of arguments, so that every conclusion “follows in one step” from its premises.
- This will be done via *natural deduction* (next chapter and couple of weeks!), based on *sequents*, which represent arguments.

Introduction

Logic is about making statements: Socrates is a goat

Logic is about making statements:

$$\overbrace{\underbrace{\text{Socrates}}_{\text{constant}}\ \underbrace{\text{is a goat}}_{\text{predicate}}}^{\text{(natural language) sentence}}$$

Logic is about making statements:

$$\overbrace{\underbrace{\text{Socrates}}_{\text{constant}}\ \underbrace{\text{is a goat}}_{\text{predicate}}}^{\text{(natural language) sentence}}$$

What’s a predicate?

- It will formally be defined later, when we deal with *predicate logics*
- A predicate relates various objects (constants) to each other, e.g.:
  - isGoat(Socrates) or isGoat(Goat)
  - isFootballer(Socrates)
  - Kicks(Socrates,Goat) or Kicks(Socrates,Ball)

Logic is about making statements: $\overbrace{\underbrace{\text{Socrates}}_{\text{constant}}\ \underbrace{\text{is a goat}}_{\text{predicate}}}^{\text{(natural language) sentence}}$

What's a predicate?

- It will formally be defined later, when we deal with *predicate logics*
- A predicate relates various objects (constants) to each other, e.g.:
  - isGoat(Socrates) or isGoat(Goat)
  - isFootballer(Socrates)
  - Kicks(Socrates,Goat) or Kicks(Socrates,Ball)
- Actually, predicates do not exist in *propositional logics*, where we have *propositions* instead, e.g.,
  - SocratesIsGoat or GoatIsGoat
  - SocratesIsFootballer
  - SocratesKicksGoat or SocratesKicksBall

  → Since this is way too long, we usually just write p, q, r, etc.

Basic Definitions: Terminology

Atoms refer to any “atomic” truth statement:

- true (denoted by $\top$, T, or 1)
- false (denoted by $\bot$, F, or 0)
- any propositional symbol (denoted by $p, q, r, \ldots$)

What’s missing for *non-atomic* statements? Connectives!

- Socrates is a goat, ...
  - because ...
  - although ...
  - until ...
  - and ...
  - or ...
- It is not true that ...
  - Socrates is a goat
  - ...

Basic Definitions: Syntax of Connectives

Which connectives do exist in propositional logic?

- ... and ...: $\wedge$ e.g., $(p \wedge \top)$ or $(p \wedge (q \wedge r))$

Basic Definitions: Syntax of Connectives

Which connectives do exist in propositional logic?

- ... and ...: $\wedge$ e.g., $(p \wedge \top)$ or $(p \wedge (q \wedge r))$
- ... or ...: $\vee$ e.g., $(\bot \vee \top)$ or $(p \vee (q \wedge r))$

Basic Definitions: Syntax of Connectives

Which connectives do exist in propositional logic?

- ... and ...: $\wedge$ e.g., $(p \wedge \top)$ or $(p \wedge (q \wedge r))$
- ... or ...: $\vee$ e.g., $(\bot \vee \top)$ or $(p \vee (q \wedge r))$
- if ..., then ...: $\rightarrow$ e.g., $(p \rightarrow q)$ or $((p \wedge q) \rightarrow (p \vee q))$
  also: ... implies ...

Basic Definitions: Syntax of Connectives

Which connectives do exist in propositional logic?

- ... and ...: $\wedge$ e.g., $(p \wedge \top)$ or $(p \wedge (q \wedge r))$
- ... or ...: $\vee$ e.g., $(\bot \vee \top)$ or $(p \vee (q \wedge r))$
- if ..., then ...: $\rightarrow$ e.g., $(p \rightarrow q)$ or $((p \wedge q) \rightarrow (p \vee q))$
  also: ... implies ...
- ... if and only if ...: $\leftrightarrow$ e.g., $(p \leftrightarrow q)$ or $((p \wedge q) \leftrightarrow (q \wedge p))$

Basic Definitions: Syntax of Connectives

Which connectives do exist in propositional logic?

- ... and ...: $\wedge$ e.g., $(p \wedge \top)$ or $(p \wedge (q \wedge r))$
- ... or ...: $\vee$ e.g., $(\bot \vee \top)$ or $(p \vee (q \wedge r))$
- if ..., then ...: $\rightarrow$ e.g., $(p \rightarrow q)$ or $((p \wedge q) \rightarrow (p \vee q))$
  also: ... implies ...
- ... if and only if ...: $\leftrightarrow$ e.g., $(p \leftrightarrow q)$ or $((p \wedge q) \leftrightarrow (q \wedge p))$
- not ...: $\neg$ e.g., $((\neg p) \rightarrow q)$ or $\neg(p \rightarrow q)$

Basic Definitions: Semantics of Connectives

What do these connectives *mean*?

- The semantics is defined in terms of truth tables.
- A truth table for a formula tells us for each interpretation of the proposition symbols whether the formula is true or false.

Basic Definitions: Semantics of Connectives

What do these connectives *mean*?

- The semantics is defined in terms of truth tables.
- A truth table for a formula tells us for each interpretation of the proposition symbols whether the formula is true or false.
- Examples:
  - $(\neg p)$ inverts p's truth value: $\top$ is switched to $\bot$, and vice versa.

Basic Definitions: Semantics of Connectives

What do these connectives *mean*?

- The semantics is defined in terms of truth tables.
- A truth table for a formula tells us for each interpretation of the proposition symbols whether the formula is true or false.
- Examples:
  - $(\neg p)$ inverts p's truth value: $\top$ is switched to $\bot$, and vice versa.
  - $(p \wedge q)$ is true if and only if both p and q are true.

Basic Definitions: Semantics of Connectives

What do these connectives *mean*?

- The semantics is defined in terms of truth tables.
- A truth table for a formula tells us for each interpretation of the proposition symbols whether the formula is true or false.
- Examples:
  - $(\neg p)$ inverts p's truth value: $\top$ is switched to $\bot$, and vice versa.
  - $(p \wedge q)$ is true if and only if both p and q are true.
  - $(p \vee q)$ is true if and only if at least one of p and q is true.

Basic Definitions: Semantics of Connectives

What do these connectives *mean*?

- The semantics is defined in terms of truth tables.
- A truth table for a formula tells us for each interpretation of the proposition symbols whether the formula is true or false.
- Examples: (expressed as *truth tables*)

p	$\neg$
$\bot$	$\top$
$\top$	$\bot$

Basic Definitions: Semantics of Connectives

What do these connectives *mean*?

- The semantics is defined in terms of truth tables.
- A truth table for a formula tells us for each interpretation of the proposition symbols whether the formula is true or false.
- Examples: (expressed as *truth tables*)

p	$\neg$
$\bot$	$\top$
$\top$	$\bot$

p	q	$\wedge$
$\bot$	$\bot$	$\bot$
$\bot$	$\top$	$\bot$
$\top$	$\bot$	$\bot$
$\top$	$\top$	$\top$

Basic Definitions: Semantics of Connectives

What do these connectives *mean*?

- The semantics is defined in terms of truth tables.
- A truth table for a formula tells us for each interpretation of the proposition symbols whether the formula is true or false.
- Examples: (expressed as *truth tables*)

p	$\neg$
$\bot$	$\top$
$\top$	$\bot$

p	q	$\wedge$
$\bot$	$\bot$	$\bot$
$\bot$	$\top$	$\bot$
$\top$	$\bot$	$\bot$
$\top$	$\top$	$\top$

p	q	$\vee$
$\bot$	$\bot$	$\bot$
$\bot$	$\top$	$\top$
$\top$	$\bot$	$\top$
$\top$	$\top$	$\top$

Basic Definitions: Semantics of Connectives

What do these connectives *mean*?

- The semantics is defined in terms of truth tables.
- A truth table for a formula tells us for each interpretation of the proposition symbols whether the formula is true or false.
- Examples: (expressed as *truth tables*)

p	$\neg$
0	1
1	0

p	q	$\wedge$
0	0	0
0	1	0
1	0	0
1	1	1

p	q	$\vee$
0	0	0
0	1	1
1	0	1
1	1	1

We will henceforth use 0/1 because its readability is so much improved!

Basic Definitions: Semantics of Connectives in *Natural Language*

Note that natural language does not always translate 1-to-1 to logics:

- "Jane and Jill went up the hill":
  says more than just WentUpHill(Jane) $\wedge$ WentUpHill(Jill),
  because it means that they went there *together*.

Basic Definitions: Semantics of Connectives in *Natural Language*

Note that natural language does not always translate 1-to-1 to logics:

- "Jane and Jill went up the hill":
  says more than just WentUpHill(Jane) ∧ WentUpHill(Jill),
  because it means that they went there *together*.
- "One false move and I will shoot!"
  Does not mean $Move(You) \wedge Shoot(I)$, but
  $Move(You) \rightarrow Shoot(I)$

Basic Definitions: Semantics of Connectives in *Natural Language*

Note that natural language does not always translate 1-to-1 to logics:

- "Jane and Jill went up the hill":
  says more than just WentUpHill(Jane) $\wedge$ WentUpHill(Jill),
  because it means that they went there *together*.
- "One false move and I will shoot!"
  Does not mean $\mathit{Move}(\mathit{You}) \wedge \mathit{Shoot}(I)$, but
  $\mathit{Move}(\mathit{You}) \rightarrow \mathit{Shoot}(I)$
- Funnily, "Don't move or I shoot":
  Is not meant to mean $\neg\mathit{Move}(\mathit{You}) \vee \mathit{Shoot}(I)$, but also means
  $\mathit{Move}(\mathit{You}) \rightarrow \mathit{Shoot}(I)$, but both are equivalent.

Basic Definitions: Semantics of Connectives in *Natural Language*

Note that natural language does not always translate 1-to-1 to logics:

- "Jane and Jill went up the hill":
  says more than just WentUpHill(Jane) $\land$ WentUpHill(Jill),
  because it means that they went there *together*.
- "One false move and I will shoot!"
  Does not mean $Move(You) \land Shoot(I)$, but
  $Move(You) \rightarrow Shoot(I)$
- Funnily, "Don't move or I shoot":
  Is not meant to mean $\neg Move(You) \lor Shoot(I)$, but also means
  $Move(You) \rightarrow Shoot(I)$, but both are equivalent.

Modeling the "real intention" behind (informal) natural language is one the learning goals of this course!

Propositional Calculus

Formal Semantics: Semantics of ¬, ∧, ∨, →

- The formal semantics of a propositional formulae is given by truth tables. (Which you already saw.)
- Recall that truth tables formally use the values ⊤ (true) and ⊥ (false), though we use 1 and 0 for better readability.
- Truth tables: (Note how they are created)

p	$\neg$

p	q	$\wedge$

p	q	$\vee$

Formal Semantics: Semantics of $\neg$, $\wedge$, $\vee$, $\rightarrow$

- The formal semantics of a propositional formulae is given by truth tables. (Which you already saw.)
- Recall that truth tables formally use the values $\top$ (true) and $\bot$ (false), though we use 1 and 0 for better readability.
- Truth tables: (Note how they are created)

p	$\neg$
0	1
1	0

p	q	$\wedge$
0	0	0
0	1	0
1	0	0
1	1	1

p	q	$\vee$
0	0	0
0	1	1
1	0	1
1	1	1

Formal Semantics: Semantics of $\neg$, $\wedge$, $\vee$, $\rightarrow$

- The formal semantics of a propositional formulae is given by truth tables. (Which you already saw.)
- Recall that truth tables formally use the values $\top$ (true) and $\bot$ (false), though we use 1 and 0 for better readability.
- Truth tables: (Note how they are created)

p	$\neg$
0	1
1	0

p	q	$\wedge$
0	0	0
0	1	0
1	0	0
1	1	1

p	q	$\vee$
0	0	0
0	1	1
1	0	1
1	1	1

p	q	$\rightarrow$
0	0	
0	1	
1	0	
1	1	

- Example implication: If the light is red (p), you must stop (q).

Formal Semantics: Semantics of $\neg$, $\wedge$, $\vee$, $\rightarrow$

- The formal semantics of a propositional formulae is given by truth tables. (Which you already saw.)
- Recall that truth tables formally use the values $\top$ (true) and $\bot$ (false), though we use 1 and 0 for better readability.
- Truth tables: (Note how they are created)

p	$\neg$
0	1
1	0

p	q	$\wedge$
0	0	0
0	1	0
1	0	0
1	1	1

p	q	$\vee$
0	0	0
0	1	1
1	0	1
1	1	1

p	q	$\rightarrow$
0	0	1
0	1	1
1	0	0
1	1	1

- Example implication: If the light is red (p), you must stop (q).

Formal Semantics: Semantics of $\leftarrow$, $\leftrightarrow$

- Some “additional” truth tables:

p	q	$\rightarrow$
0	0	1
0	1	1
1	0	0
1	1	1

p	q	$\leftarrow$
0	0	1
0	1	0
1	0	1
1	1	1

Formal Semantics: Semantics of $\leftarrow$, $\leftrightarrow$

- Some “additional” truth tables:

p	q	$\rightarrow$
0	0	1
0	1	1
1	0	0
1	1	1

p	q	$\leftarrow$
0	0	1
0	1	0
1	0	1
1	1	1

p	q	$\leftrightarrow$
0	0	
0	1	
1	0	
1	1	

Formal Semantics: Semantics of $\leftarrow$, $\leftrightarrow$

- Some “additional” truth tables:

p	q	$\rightarrow$
0	0	1
0	1	1
1	0	0
1	1	1

p	q	$\leftarrow$
0	0	1
0	1	0
1	0	1
1	1	1

p	q	$\leftrightarrow$
0	0	1
0	1	0
1	0	0
1	1	1

- We will not need them since we restrict to the standard connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$.

Formal Semantics: Expressing Arbitrary Formulae with Truth Tables

Truth tables can be used to express arbitrary formulae, e.g.,

$p \wedge \neg q$

-

p	q	$\neg q$	$\wedge$
0	0		
0	1		
1	0		
1	1		

Formal Semantics: Expressing Arbitrary Formulae with Truth Tables

Truth tables can be used to express arbitrary formulae, e.g.,

$p \land \neg q$

p	q	$\neg q$	$\land$
0	0	1	
0	1	0	
1	0	1	
1	1	0	

Formal Semantics: Expressing Arbitrary Formulae with Truth Tables

Truth tables can be used to express arbitrary formulae, e.g.,

$p \land \neg q$

p	q	$\neg q$	$\land$
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	0

Formal Semantics: Expressing Arbitrary Formulae with Truth Tables

Truth tables can be used to express arbitrary formulae, e.g.,

$p \wedge \neg q$

p	q	$\neg q$	$\wedge$
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	0

$\neg p \vee q$

p	q	$\neg p$	$\neg p \vee q$
0	0		
0	1		
1	0		
1	1		

Formal Semantics: Expressing Arbitrary Formulae with Truth Tables

Truth tables can be used to express arbitrary formulae, e.g.,

$p \wedge \neg q$

p	q	$\neg q$	$\wedge$
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	0

$\neg p \vee q$

p	q	$\neg p$	$\neg p \vee q$
0	0	1	
0	1	1	
1	0	0	
1	1	0	

Formal Semantics: Expressing Arbitrary Formulae with Truth Tables

Truth tables can be used to express arbitrary formulae, e.g.,

- $p \wedge \neg q$

p	q	$\neg q$	$\wedge$
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	0

$\neg p \vee q$

p	q	$\neg p$	$\neg p \vee q$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1

Formal Semantics: Expressing Arbitrary Formulae with Truth Tables

Truth tables can be used to express arbitrary formulae, e.g.,

$p \land \neg q$

p	q	$\neg q$	$\land$
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	0

$\neg p \lor q$ that's $p \rightarrow q$!

p	q	$\neg p$	$\neg p \lor q$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1

Formal Semantics: Expressing Arbitrary Formulae with Truth Tables

Truth tables can be used to express arbitrary formulae, e.g.,

- $p \land \neg q$

p	q	$\neg q$	$\land$
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	0

$\neg p \lor q$ that's $p \rightarrow q$!

p	q	$\neg p$	$\neg p \lor q$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1

- $p \rightarrow (q \rightarrow p)$

p	q	$q \rightarrow p$	$p \rightarrow (q \rightarrow p)$
0	0		
0	1		
1	0		
1	1		

Formal Semantics: Expressing Arbitrary Formulae with Truth Tables

Truth tables can be used to express arbitrary formulae, e.g.,

- $p \wedge \neg q$

p	q	$\neg q$	$\wedge$
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	0

$\neg p \vee q$ that's $p \rightarrow q$!

p	q	$\neg p$	$\neg p \vee q$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1

- $p \rightarrow (q \rightarrow p)$

p	q	$q \rightarrow p$	$p \rightarrow (q \rightarrow p)$
0	0	1	
0	1	0	
1	0	1	
1	1	1	

Such a formula, which always evaluates to true is called a tautology.

Formal Semantics: Expressing Arbitrary Formulae with Truth Tables

Truth tables can be used to express arbitrary formulae, e.g.,

- $p \wedge \neg q$

p	q	$\neg q$	$\wedge$
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	0

$\neg p \vee q$ that's $p \rightarrow q$!

p	q	$\neg p$	$\neg p \vee q$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1

- $p \rightarrow (q \rightarrow p)$

p	q	$q \rightarrow p$	$p \rightarrow (q \rightarrow p)$
0	0	1	1
0	1	0	1
1	0	1	1
1	1	1	1

Such a formula, which always evaluates to true is called a tautology.

Formal Semantics: Interpretations and Properties of Formulae

Definition:
An *Interpretation* of a formula ϕ, defined over a set P of propositional symbols is an assignment of truth values to symbols in P.

Example:

- Let $p = \mathit{LogicIsInteresting}$
- Let $q = \mathit{PascalsSlidesAreWellDesigned}$
- Let $r = \mathit{studentsUnderstandContent}$
- Now consider $(p \wedge q) \rightarrow r$

p	q	r	$(p \wedge q) \rightarrow r$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Here, an interpretation could be:
$I(p) = 1$, $I(q) = 1$, $I(r) = 0$

This interpretation does not make the formula true!
(Interpretations can be thought of as rows in the table)

Formal Semantics: Interpretations and Properties of Formulae, cont'd

Why do we need interpretations?

Because now we can define important properties!

- A formula ϕ is a ***tautology*** if:

Formal Semantics: Interpretations and Properties of Formulae, cont'd

Why do we need interpretations?
Because now we can define important properties!

- A formula ϕ is a ***tautology*** if:
  ϕ is true under every interpretation.

Formal Semantics: Interpretations and Properties of Formulae, cont'd

Why do we need interpretations?
Because now we can define important properties!

- A formula ϕ is a ***tautology*** if:
  ϕ is true under every interpretation.
- A formula ϕ is ***satisfiable*** if:

Formal Semantics: Interpretations and Properties of Formulae, cont'd

Why do we need interpretations?
Because now we can define important properties!

- A formula ϕ is a ***tautology*** if:
  ϕ is true under every interpretation.
- A formula ϕ is ***satisfiable*** if:
  There exists an interpretation that makes ϕ true.

Formal Semantics: Interpretations and Properties of Formulae, cont'd

Why do we need interpretations?
Because now we can define important properties!

- A formula ϕ is a ***tautology*** if:
  ϕ is true under every interpretation.
- A formula ϕ is ***satisfiable*** if:
  There exists an interpretation that makes ϕ true.
- A formula ϕ is ***unsatisfiable*** if:

Formal Semantics: Interpretations and Properties of Formulae, cont'd

Why do we need interpretations?
Because now we can define important properties!

- A formula ϕ is a ***tautology*** if:
  ϕ is true under every interpretation.
- A formula ϕ is ***satisfiable*** if:
  There exists an interpretation that makes ϕ true.
- A formula ϕ is ***unsatisfiable*** if:
  There does not exist an interpretation that makes ϕ true.
  Or equivalently: If ϕ is false under every interpretation.

Formal Semantics: Interpretations and Properties of Formulae, cont'd

Why do we need interpretations?
Because now we can define important properties!

- A formula ϕ is a ***tautology*** if:
  ϕ is true under every interpretation.
- A formula ϕ is ***satisfiable*** if:
  There exists an interpretation that makes ϕ true.
- A formula ϕ is ***unsatisfiable*** if:
  There does not exist an interpretation that makes ϕ true.
  Or equivalently: If ϕ is false under every interpretation.

So, could we say "Formula ϕ is true"?

Formal Semantics: Interpretations and Properties of Formulae, cont'd

Why do we need interpretations?
Because now we can define important properties!

- A formula ϕ is a ***tautology*** if:
  ϕ is true under every interpretation.
- A formula ϕ is ***satisfiable*** if:
  There exists an interpretation that makes ϕ true.
- A formula ϕ is ***unsatisfiable*** if:
  There does not exist an interpretation that makes ϕ true.
  Or equivalently: If ϕ is false under every interpretation.

So, could we say "Formula ϕ is true"?
Not really... Only that it is true under a certain interpretation.

Syntax Simplifications: Precedence of Connectives

Our connectives use some precedence, which we exploit to eliminate parentheses! Connectives, ordered by precedence:

- Highest: $\neg$ e.g., $\neg p \rightarrow q \equiv (\neg p) \rightarrow q$

Syntax Simplifications: Precedence of Connectives

Our connectives use some precedence, which we exploit to eliminate parentheses! Connectives, ordered by precedence:

- Highest: $\neg$ e.g., $\neg p \rightarrow q \equiv (\neg p) \rightarrow q$
- Second-highest: $\wedge$ e.g., $p \wedge q \vee r \equiv (p \wedge q) \vee r$

Syntax Simplifications: Precedence of Connectives

Our connectives use some precedence, which we exploit to eliminate parentheses! Connectives, ordered by precedence:

- Highest: $\neg$ e.g., $\neg p \rightarrow q \equiv (\neg p) \rightarrow q$
- Second-highest: $\wedge$ e.g., $p \wedge q \vee r \equiv (p \wedge q) \vee r$
- Mid: $\vee$ e.g., $p \rightarrow q \vee r \equiv p \rightarrow (q \vee r)$

Syntax Simplifications: Precedence of Connectives

Our connectives use some precedence, which we exploit to eliminate parentheses! Connectives, ordered by precedence:

- Highest: $\neg$ e.g., $\neg p \rightarrow q \equiv (\neg p) \rightarrow q$
- Second-highest: $\wedge$ e.g., $p \wedge q \vee r \equiv (p \wedge q) \vee r$
- Mid: $\vee$ e.g., $p \rightarrow q \vee r \equiv p \rightarrow (q \vee r)$
- Second-Lowest: $\rightarrow$ e.g., $p \rightarrow \neg q \leftrightarrow r \equiv (p \rightarrow (\neg q)) \leftrightarrow r$

Syntax Simplifications: Precedence of Connectives

Our connectives use some precedence, which we exploit to eliminate parentheses! Connectives, ordered by precedence:

- Highest: $\neg$ e.g., $\neg p \rightarrow q \equiv (\neg p) \rightarrow q$
- Second-highest: $\wedge$ e.g., $p \wedge q \vee r \equiv (p \wedge q) \vee r$
- Mid: $\vee$ e.g., $p \rightarrow q \vee r \equiv p \rightarrow (q \vee r)$
- Second-Lowest: $\rightarrow$ e.g., $p \rightarrow \neg q \leftrightarrow r \equiv (p \rightarrow (\neg q)) \leftrightarrow r$
- Lowest: $\leftrightarrow$ e.g., $\neg p \vee q \leftrightarrow q \wedge r \equiv ((\neg p) \vee q) \leftrightarrow (q \wedge r)$

Syntax Simplifications: Precedence of Connectives

Our connectives use some precedence, which we exploit to eliminate parentheses! Connectives, ordered by precedence:

- Highest: $\neg$ e.g., $\neg p \rightarrow q \equiv (\neg p) \rightarrow q$
- Second-highest: $\wedge$ e.g., $p \wedge q \vee r \equiv (p \wedge q) \vee r$
- Mid: $\vee$ e.g., $p \rightarrow q \vee r \equiv p \rightarrow (q \vee r)$
- Second-Lowest: $\rightarrow$ e.g., $p \rightarrow \neg q \leftrightarrow r \equiv (p \rightarrow (\neg q)) \leftrightarrow r$
- Lowest: $\leftrightarrow$ e.g., $\neg p \vee q \leftrightarrow q \wedge r \equiv ((\neg p) \vee q) \leftrightarrow (q \wedge r)$

We reduce parentheses to simplify and avoid confusion by exploiting:

- *precedence*, e.g., we write: $\neg p \rightarrow q$ instead of $((\neg p) \rightarrow q)$

Syntax Simplifications: Precedence of Connectives

Our connectives use some precedence, which we exploit to eliminate parentheses! Connectives, ordered by precedence:

- Highest: $\neg$ e.g., $\neg p \rightarrow q \equiv (\neg p) \rightarrow q$
- Second-highest: $\wedge$ e.g., $p \wedge q \vee r \equiv (p \wedge q) \vee r$
- Mid: $\vee$ e.g., $p \rightarrow q \vee r \equiv p \rightarrow (q \vee r)$
- Second-Lowest: $\rightarrow$ e.g., $p \rightarrow \neg q \leftrightarrow r \equiv (p \rightarrow (\neg q)) \leftrightarrow r$
- Lowest: $\leftrightarrow$ e.g., $\neg p \vee q \leftrightarrow q \wedge r \equiv ((\neg p) \vee q) \leftrightarrow (q \wedge r)$

We reduce parentheses to simplify and avoid confusion by exploiting:

- *precedence*, e.g., we write: $\neg p \rightarrow q$ instead of $((\neg p) \rightarrow q)$
- *associativity*, e.g., we write:
  - $p \wedge q \wedge r$ instead of $(p \wedge (q \wedge r))$
  - $(p \wedge \neg q \wedge r) \rightarrow (p \vee \neg q \vee r)$ instead of $(((p \wedge (\neg q)) \wedge r) \rightarrow (p \vee ((\neg q) \vee r)))$

Connective Scopes and Main Connective: Connective Scopes

- Every connective has a *scope*.
- “[The scope of a connective] is defined to be the shortest formula or subformula in which that occurrence lies.” (Logic Notes)
- Examples: In the formula $\neg(p \wedge q) \rightarrow ((p \vee r) \rightarrow \neg s)$
  - ... the scope of its first $\neg$ is $(p \wedge q)$
  - ... the scope of its second $\neg$ is s

Connective Scopes and Main Connective: Main Connective

- Every formula has a *main connective*:
- “[T]he main connective of any formula [...] is the one which is not inside the scope of any other. [...] The scope of the main connective is the whole formula.” (Logic Notes)
- Examples: The main connective of . . .
    - $\neg(p \land q) \rightarrow ((p \lor r) \rightarrow \neg s)$ is:

Connective Scopes and Main Connective: Main Connective

- Every formula has a *main connective*:
- "[T]he main connective of any formula [...] is the one which is not inside the scope of any other. [...] The scope of the main connective is the whole formula." (Logic Notes)
- Examples: The main connective of . . .
  - $\neg(p \land q) \rightarrow ((p \lor r) \rightarrow \neg s)$ is: the first →

Connective Scopes and Main Connective: Main Connective

- Every formula has a *main connective*:
- “[T]he main connective of any formula [...] is the one which is not inside the scope of any other. [...] The scope of the main connective is the whole formula.” (Logic Notes)
- Examples: The main connective of . . .
  - $\neg(p \land q) \to ((p \lor r) \to \neg s)$ is: the first $\to$
  - $(p \land q) \lor r$ is:

Connective Scopes and Main Connective: Main Connective

- Every formula has a *main connective*:
- “[T]he main connective of any formula [...] is the one which is not inside the scope of any other. [...] The scope of the main connective is the whole formula.” (Logic Notes)
- Examples: The main connective of . . .
  - $\neg(p \wedge q) \rightarrow ((p \vee r) \rightarrow \neg s)$ is: the first $\rightarrow$
  - $(p \wedge q) \vee r$ is: $\vee$

Connective Scopes and Main Connective: Main Connective

- Every formula has a *main connective*:
- “[T]he main connective of any formula [...] is the one which is not inside the scope of any other. [...] The scope of the main connective is the whole formula.” (Logic Notes)
- Examples: The main connective of . . .
  - $\neg(p \wedge q) \rightarrow ((p \vee r) \rightarrow \neg s)$ is: the first $\rightarrow$
  - $(p \wedge q) \vee r$ is: $\vee$
  - What’s the main connective of $(p \wedge q) \vee r \vee (q \rightarrow r)$?

Connective Scopes and Main Connective: Main Connective

- Every formula has a *main connective*:
- "[T]he main connective of any formula [...] is the one which is not inside the scope of any other. [...] The scope of the main connective is the whole formula." (Logic Notes)
- Examples: The main connective of . . .
  - $\neg(p \wedge q) \rightarrow ((p \vee r) \rightarrow \neg s)$ is: the first $\rightarrow$
  - $(p \wedge q) \vee r$ is: $\vee$
  - What's the main connective of $(p \wedge q) \vee r \vee (q \rightarrow r)$?
    Recall that "$(p \wedge q) \vee r \vee (q \rightarrow r)$" is only syntactic sugar!
    - It was either $((p \wedge q) \vee r) \vee (q \rightarrow r)$ [then, it's the right $\vee$],
    - or it was $(p \wedge q) \vee (r \vee (q \rightarrow r))$ [then, it's the left $\vee$]

    → Formally, assiciativity defines *uniquely* what a formula with missing parentheses defines. (But that's not important for this course.)

Connective Scopes and Main Connective: Main Connective

- Every formula has a *main connective*:
- "[T]he main connective of any formula [...] is the one which is not inside the scope of any other. [...] The scope of the main connective is the whole formula." (Logic Notes)
- Examples: The main connective of . . .
  - $\neg(p \land q) \rightarrow ((p \lor r) \rightarrow \neg s)$ is: the first $\rightarrow$
  - $(p \land q) \lor r$ is: $\lor$
  - What's the main connective of $(p \land q) \lor r \lor (q \rightarrow r)$?
    Recall that "$(p \land q) \lor r \lor (q \rightarrow r)$" is only syntactic sugar!
    - It was either $((p \land q) \lor r) \lor (q \rightarrow r)$ [then, it's the right $\lor$],
    - or it was $(p \land q) \lor (r \lor (q \rightarrow r))$ [then, it's the left $\lor$]

    → Formally, assiciativity defines *uniquely* what a formula with missing parentheses defines. (But that's not important for this course.)
- Why is it important to identify the main connective?
  Because the main connective defines the "type" of the formula, which defines what we are allowed to do in our proofs.

Connective Scopes and Main Connective: Type of Formula

The main connective dictates the type of a formula:

- if main connective is $\neg$, formula is a *negation*
- ... $\wedge$, ... *conjunction*
- ... $\vee$, ... *disjunction*
- ... $\rightarrow$, ... *implication*
- ... $\leftrightarrow$, ... *double-implication*

Substitution: Substitutions of Formulae

What is a substitution?

- “Formula A is a substitution instance of formula B if and only if A results from B by substitution of formulas for sentence letters.” (Logic Notes) – (Definition is specific to *propositional* logic.)

Substitution: Substitutions of Formulae

What is a substitution?

- “Formula A is a substitution instance of formula B if and only if A results from B by substitution of formulas for sentence letters.” (Logic Notes) – (Definition is specific to *propositional* logic.)

Example:

- “Original” formula: $q \lor p$
- One of its substitution instances is $(p \land q) \lor \neg r$, because:
  - q got substituted by $(p \land q)$
  - p got substituted by $\neg r$

Substitution: Substitutions of Formulae

What is a substitution?

- “Formula A is a substitution instance of formula B if and only if A results from B by substitution of formulas for sentence letters.” (Logic Notes) – (Definition is specific to *propositional* logic.)

Non-Example:

- “Original” formula: $q \vee q$
- The formula $(p \wedge q) \vee \neg r$ is *not a* substitution instance of it (because the left part had to be the same as the right)

Summary

Content of this Lecture

- Organizational Matters
- Introduction to *Propositional Logic*
  - Its syntax and semantics, interpretations
  - What connectives exist (and which don't)
  - How to identify the type of a formula (e.g., negation, conjunction, disjunction, implication, double-implication)
  - What's a substitution

→ Logic Notes sections:

- Complete *1. Introduction* except *Inference in the abstract*
- *3. More about propositional logic*: *Truth tables*.